Listing of the Claims:

The following is a complete listing of all the claims in the application, with an indication of the status of each:

1. (Currently amended) A method of IP fragmentation on a network processor comprising the steps of:

storing ~~the~~ a frame to be fragmented in a series of buffers, chained together by a linked list;

associating a buffer control block with each buffer;

associating a frame control block with each frame;

receiving frames into a queue to await dispatch to a network processor;

associating a queue control block with a queue of frames to be transmitted;

assigning additional buffers and additional frame control blocks for each multicast target and linking these additional frame control blocks with the original frame control block associated with the frame;

using a multicast counter to determine when all frame fragments have been transmitted;

returning buffers and frame control blocks assigned to each fragment toa ~~the~~ free queue as each fragment is transmitted;

and

returning the original buffers and frame control block to the free queues after the frame has been sent to all of the multicast targets.

2. (Currently amended) The method for IP fragmentation as recited in claim 1, wherein the buffer control block associated with each buffer forms a linked list for chaining buffers into a frame and contains a plurality of fields, including separate fields to

store a pointer to a ~~the~~ next buffer in the frame;

store an ~~the~~ offset of a ~~the~~ first valid byte of data in the next buffer of a frame;

store an ~~the~~ offset of a ~~the~~ last valid byte of data in the next buffer of a frame; and indicate whether the next buffer in the frame should be returned to the free buffer or queue or retained so as to continue multicast transmission.

3. (Currently amended) The method for IP fragmentation as recited in claim 1, wherein the frame control block associated with each frame forms a linked list for chaining frames into a queue and contains a plurality of fields, including separate fields to

store a pointer to a ~~the~~ next frame in the queue;

store a count of a ~~the~~ total number of bytes of the next frame in the queue;

store an ~~the~~ address of a ~~the~~ first buffer in a frame;

store a ~~the~~ starting byte position of valid data in the first buffer of a frame;

store an ~~the~~ ending byte position of valid data in the first buffer of a frame; and

store information on a ~~the~~ format and a ~~the~~ type of a ~~the~~ frame to be transmitted.

4. (Currently amended) The method for IP fragmentation as recited in claim 1, wherein the step of receiving frames into a queue comprises the further steps of:

popping a free buffer address from a ~~the~~ head of the free buffer queue;

popping a free frame control block from the head of the free frame control block queue;

writing frame data to a ~~the~~ buffer;

writing control information, including a ~~the~~ first buffer address, ~~the~~ starting and ending byte positions for valid data in the first buffer, to the frame control block;

setting a working byte count register to a ~~the~~ number of bytes written to the first buffer;

repeating this process until an ~~the~~ entire frame is written to buffers; and

adding a ~~the~~ frame to a ~~the~~ tail of an input queue to await dispatch to the network processor.

5. (Currently amended) The method for IP fragmentation as recited in claim 1,

wherein the queue control block associated with the queue of frames to be transmitted includes a plurality of fields, including separate fields to

store an ~~the~~ address of the frame control block associated with the frame at a ~~the~~ head of the queue;

store a count of a ~~the~~ total number of valid bytes in the frame at a ~~the~~ top of the queue; and

store the address of the frame control block associated with the frame at a ~~the~~ tail of the queue.

6. (Currently amended) A method for IP reassembly on a network processor comprising the steps of:

storing frames to be reassembled in a series of buffers chained together by a linked list;

associating a buffer control block with each buffer;

associating a frame control block with each frame;

receiving frames into a queue to await dispatch to a network processor;

associating two or more received frames that are to be reassembled into a single transmit frame;

updating the buffer control blocks to modify the linked list of buffers for combining multiple frames into a single reassembled frame;

associating a frame control block with the reassembled frame;

returning unused frame control blocks to a free queue;

associating a queue control block with a queue of frames to be transmitted; and

returning the buffers and frame control block of the reassembled frame to a ~~the~~ free queue as the frame is transmitted.

7. (Currently Amended) A network processor for processing IP fragmentation comprising:

means for storing a ~~the~~ frame to be fragmented in a series of buffers, chained together by a linked list;

means for associating a buffer control block with each buffer and associating a frame control block with each frame;

means for receiving frames into a queue to await dispatch to a network processor;

means for associating a queue control block with a queue of frames to be transmitted;

means for assigning additional buffers and additional frame control blocks for each multicast target and linking these additional frame control blocks with the original frame control block associated with the frame;

means using a multicast counter for determining when all frame fragments have been transmitted; and

means for returning buffers and frame control blocks assigned to each fragment to a ~~the~~ free queue as each fragment is transmitted and returning the original buffers and frame control block to ~~the~~ free queues after the frame has been sent to all of the multicast targets.

8. (Currently amended) The network processor as recited in claim 7, wherein the buffer control block associated with each buffer forms a linked list for chaining buffers into a frame and contains a plurality of fields, including separate fields to

store a pointer to a ~~the~~ next buffer in the frame;

store an ~~the~~ offset of a ~~the~~ first valid byte of data in the next buffer of a frame;

store an ~~the~~ offset of a ~~the~~ last valid byte of data in the next buffer of a frame; and

indicate whether the next buffer in the frame should be returned to the free buffer or queue or retained so as to continue multicast transmission.

9. (Currently amended) The network processor as recited in claim 7, wherein the frame control block associated with each frame forms a linked list for chaining frames into a queue and contains a plurality of fields, including separate fields to

store a pointer to a ~~the~~ next frame in the queue;

store a count of a ~~the~~ total number of bytes of the next frame in the queue;

store an ~~the~~ address of a ~~the~~ first buffer in a frame;

store a ~~the~~ starting byte position of valid data in the first buffer of a frame;

store an ~~the~~ ending byte position of valid data in the first buffer of a frame; and

store information on a ~~the~~ format and a ~~the~~ type of a ~~the~~ frame to be transmitted.

10. (Currently amended) The network processor as recited in claim 7, wherein the means for receiving frames into a queue comprises:

means for popping a free buffer address from a ~~the~~ head of the free buffer queue;

means for popping a free frame control block from a ~~the~~ head of the free frame control block queue;

means for writing frame data to a ~~the~~ buffer and writing control information, including a ~~the~~ first buffer address, ~~the~~ starting and ending byte positions for valid data in the first buffer, to the frame control block;

means for setting a working byte count register to a ~~the~~ number of bytes written to the first buffer; and

means, responsive to an ~~the~~ entire frame being ~~is~~ written to buffers, for adding a ~~the~~ frame to a ~~the~~ tail of an input queue to await dispatch to the network processor.

11. (Currently amended) The network processor as recited in claim 7, wherein the queue control block associated with the queue of frames to be transmitted includes a plurality of fields, including separate fields to

store an ~~the~~ address of the frame control block associated with the frame at a ~~the~~ head of the queue;

store a count of a ~~the~~ total number of valid bytes in the frame at a ~~the~~ top of the queue; and

store the address of the frame control block associated with the frame at a ~~the~~ tail of the queue.

12. (Currently amended) A network processor for IP reassembly comprising:

means for storing frames to be reassembled in a series of buffers chained together by a linked list;

means for associating a buffer control block with each buffer and associating a frame control block with each frame;

means for receiving frames into a queue to await dispatch to a network processor;

means for associating two or more received frames that are to be reassembled into a single transmit frame;

means for updating the buffer control blocks to modify the linked list of buffers for combining multiple frames into a single reassembled frame and associating a frame control block with the reassembled frame; and

means for returning unused frame control blocks to a free queue, associating a queue control block with a queue of frames to be transmitted, and returning the buffers and frame control block of the reassembled frame to a ~~the~~ free queue as the frame is transmitted.